Refatoração de Testes e Detecção de Testes Smell

Disciplina: Teste de Software
Trabalho: Análise de Eficácia de Testes com Teste de Mutação
Nome: Matheus Dias Mendes
Matrícula:
Data: 02/11/2025

Análise de Smells

- O primeiro Smell que eu encontrei foi o Convidado Misterioso, o problema é que o dado fica dependendo de dados externos cujo o conteúdo não fica visível dentro do teste, o risco é porque se o valor mudar os testes podem falhar de forma inesperada
- O segundo Smell que eu encontrei foi Teste Condicional, o problema é que os testes devem ser determinísticos e lineares, o risco seria porque partes do teste podem não ser executadas dependendo da condição, o deixando fragil
- O terceiro Smell que eu encontrei foi Teste Frágil, o problema dele é que o teste fica acoplado a implementação e não ao comportamento, o risco seria se pequenas alterações na formatação forem feitas podem quebrar o teste mesmo estando correto

Processo de Refatoração

O teste mais problemático foi o teste que utilizou o try catch

Antes:

```
  test('deve falhar ao criar usuário menor de idade', () => {
    try {
      userService.createUser('Menor', 'menor@email.com', 17);
    } catch (e) {
      expect(e.message).toBe('O usuário deve ser maior de idade.');
    }
  });
```

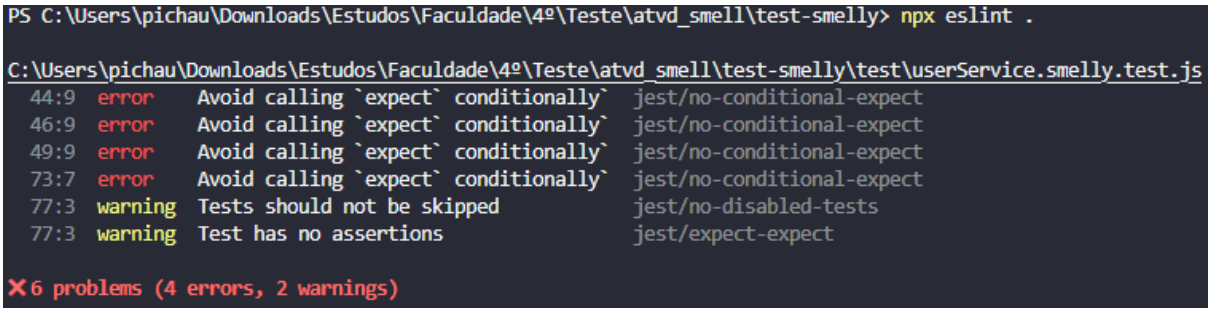
Depois:

```
  test("deve lançar erro ao tentar criar usuário menor de idade", () =>
{
    const nome = "joao";
    const email = "joao@gmail.com";
    const idade = 16;
    expect(() => {
      userService.createUser(nome, email, idade);
    }).toThrow("O usuário deve ser maior de idade");
  });
```

Eu fiz essas alterações, porque o try catch tinha o problema que se nenhuma exceção for lançada o código nunca entra no catch então eu criei um usuário dentro

do expect com a idade menor de 18 e pus o toThrow para pegar o erro de usuário com idade menor de 18

Relatório da Ferramenta



Para validar os testes utilizando ESList tive que fazer a instalação de uma versão anterior dele pois a partir da versão 9 do ESList, o formato .eslintrc.* (como .eslintrc.json, .eslintrc.js, etc.) foi descontinuado, então eu voltei para a versão 8.57.0. A execução do codigo para verificar os erros foi npx eslint test/userService.smelly.test.js.
A ferramenta detectou erro de lógica condicional dentro do teste, o aviso do teste que era para ser criado e estava com o .skip e que não tinha nenhum expect() nesse mesmo teste

Conclusão

Como conclusão eu percebi que com a refatoração, aumentou muito a clareza e a robustez da suíte de testes, e que com o uso do ESList junto com o plugin jest automatizou muito a detecção de erros no código e na lógica dos testes, o que facilita na manutenção e reduzido o risco de regressões.